AMENDMENTS TO THE CLAIMS

The following listing of claims will replace all prior versions and listings of claims in the application.

LISTING OF CLAIMS

1-7. (Cancelled).

8. (Currently Amended) A dispatching method for polling device data, comprising the steps of~~The method of Claim 7, wherein the step of polling the at least one module through the current polling task queue comprises~~:

sorting managed devices according to their types;

sorting various types of data of each managed device to different modules, wherein a priority attribute and a polling period attribute are assigned to each module respectively, wherein the polling period attribute of a module is a polling interval multiple, equal to multiple of a polling interval;

generating a data structure for describing a device type, wherein a second set of data items for describing the data structure comprises a device type, a module ID, a priority, the polling interval multiple and a corresponding daemon ID;

reading device type description data;

determining a first set of devices to be polled from the managed devices, wherein the first set of devices at least comprises an operation device set;

when a system polling is initiated, dispatching a periodical polling by determining at least one module to be polled currently from the first set of devices according to the priority attribute and the polling period attribute of the first set of devices, wherein a first

set of data items for describing the at least one module at least comprises a device ID, a module ID task ID, an occupied flag, an activation time and a priority, said activation time is the current time when inserting a task and is updated when a report about executing situation of the task sent from a daemon has been received; said occupied flag is set free after a corresponding message showing the task has been completed is received or the polling task is overtime;

inserting the at least one module to a current polling task queue according to said periodical polling;

~~a.~~ setting a polling initiating time at the summation of the current time plus ~~a~~ the polling interval, wherein the periodical polling is implemented based on said polling initiating time plus a polling interval;

~~b.~~ determining whether there is a free task in the current polling task queue based on the occupied flag; if so, continuing the process, otherwise returning to the step of determining whether there is a free task~~step b~~;

~~c.~~ selecting a next device module to be polled from the ~~current~~ operation device set; ~~and~~

~~d.~~ determining whether the information obtained by selecting a next device module to be polled~~in step c~~ is Null; if not, assigning a task ID to the selected device module and inserting the task ID into the current polling task queue, and simultaneously sending a message for initiating the polling of said device module to the corresponding daemon process, then returning to the step of determining whether there is a free task~~step b~~; if so, determining whether all tasks in the current polling task queue are in

free state, if all tasks are in free state, ending the process, otherwise returning to the step of determining whether there is a free task;~~step b.~~

providing a third set of data items for describing the first set of devices including a device type and a last polling time;

determining a second set of devices whose connection states need to be detected from the managed devices, wherein the second set of devices at least comprises a display device set; and

providing a fourth set of data items for describing the second set of devices at least comprises a device ID and a connection state.

9. (Currently Amended) The method of Claim 8, wherein the step of selecting a next device module to be polled from the ~~currently~~ operation device set ~~further~~ comprisinges:

c1. selecting the next device module;

c2. determining whether [(the current time – the last polling time)/polling interval multiple of the module] is greater than or equal to the system polling interval, if so, continuing the process, otherwise going to step c4; and

c3. determining whether there is a module with higher priority of the same device being polled in the current polling task queue; if so, returning to step c1, otherwise returning the device module information and ending step c; or

c4. determining whether said polling interval multiple is greater than one; if so, returning to step c1, otherwise returning a message of NULL and ending step c.

10. (Cancelled).